

**System and Method for Conducting A Secure Interactive Communication Session**  
**(A-70557/RMA)**

**WE CLAIM:**

5 1. A computer program product for use in conjunction with a computer system having a server and a client, the computer program product comprising a computer readable storage medium and a computer program mechanism embedded therein, the computer program mechanism, comprising: a program module that directs the computer system and/or components thereof including at least one or the client or server, to function in a specified manner to provide message communications, the message  
10 communications occurring in a computer system hardware architecture neutral and operating system neutral and network transport protocol neutral manner for secure interactive communication sessions, the program module including instructions for:

A. sending to a server, by a client, a first message containing a Client-Nonce;

B. receiving said first message including said Client-Nonce by said server;

15 C. sending to the client, by the server in response to said received first message and Client-Nonce, a second message containing a copy of the Client-Nonce extracted from the first message, and a value in the form of a Server-Nonce that was chosen by the Server that is not predictable by the Client and is unlikely to have been previously chosen by the Server; the first message and second message having substantially the same content, format and cryptographic processing;

20 D. exchanging third and fourth messages between the client and the server (client to server message) and the server and the client (server to client message) respectively, where the order that said third and fourth messages are sent and received is not material; said third and fourth messages including a content portion that is substantially the same though not necessarily identical and having substantially the same format and cryptographic processing as each other and as with subsequent data transfer  
25 messages; the data contents portions of the third and fourth message include a cryptographic transformation of at least the Client-Nonce and Server-Nonce, where the cryptographic transformation is slightly different in the third and fourth messages; and

30 E. each of the server and client examining the respective received third and fourth messages to confirm that they have the expected contents and thus were created by an entity that knew both the Client-Nonce and the Server-Nonce.

2. A hardware architecture neutral and operating system neutral and network transport neutral method for secure interactive communication sessions using less software code and network bandwidth than conventional systems, said method comprising:

35 A. sending to a server, by a client, a first message containing a Client-Nonce;

B. receiving said first message including said Client-Nonce by said server;

C. sending to the client, by the server in response to said received first message and Client-Nonce, a second message containing a copy of the Client-Nonce extracted from the first message, and a value in the form of a Server-Nonce that was chosen by the Server that is not predictable by the Client and is

unlikely to have been previously chosen by the Server; the first message and second message having substantially the same content, format and cryptographic processing;

5 D. exchanging third and fourth messages between the client and the server (client to server message) and the server and the client (server to client message) respectively, where the order that said third and fourth messages are sent and received is not material; said third and fourth messages including a content portion that is substantially the same though not necessarily identical and having substantially the same format and cryptographic processing as each other and as with subsequent data transfer messages; the data contents portions of the third and fourth message include a cryptographic transformation of at least the Client-Nonce and Server-Nonce, where the cryptographic transformation is slightly different in the third and fourth messages; and

10 E. each of the server and client examining the respective received third and fourth messages to confirm that they have the expected contents and thus were created by an entity that knew both the Client-Nonce and the Server-Nonce.

15 3. The method in claim 2, further comprising after said sever and said client have examined and confirmed that the third and fourth messages were created by entities that knew both the Client-Nonce and the Server-Nonce; F. the Client and Server optionally sending subsequent data messages that have substantially the same format and cryptographic processing as the third and fourth messages.

20 4. The method in claim 2, further comprising after a last message has been communicated between said client and said server or between said server and said client; (G) terminating the session without a separate session termination message by closing the underlying network connection.

25 5. The method in claim 3, further comprising after a last message has been communicated between said client and said server or between said server and said client, (G) terminating the session without a separate session termination message by closing the underlying network connection.

30 6. The method in claim 4, wherein the underlying network connection is a TCP based connection, by closing the TCP socket.

7. The method in claim 5, wherein the underlying network connection is a TCP based connection, by closing the TCP socket.

35 8. The method in claim 2, wherein the first and second message have no cryptographic processing when the protocol used for the messages is attempting to reuse one or more cryptographic master keys that were established in a previous messaging session, and the first and second messages have substantially the same format, and the Server verifies the existence of a Key-ID from the first message in a server cache of pairs of Key-ID and Master Key values.

9. The method in claim 8, wherein the first and second message have a common header that includes fields for Type, Version, and Content-Length; the first message contents containing a Key-ID and a Client-Nonce; and the second message contents containing the same Key-ID, the same Client-Nonce, and a new Server-Nonce.

5

10. The method in claim 8, wherein the Key-ID is a cryptographic hash of a previously set up Master Key.

11. The method in claim 10, wherein the cryptographic hash is a MD5 based hash, a SHA-1 based hash, or a SHA-256 based hash.

10

12. The method in claim 2, wherein the Client-Nonce and Server-Nonce have the same length.

13. The method in claim 2, wherein the Client-Nonce and the Server-Nonce have a length of 8 bytes, 10 bytes, 16 bytes, 20 bytes, 24 bytes, 32 bytes, 64 bytes, 96 bytes, or 128 bytes.

15

14. The method in claim 2, wherein the first and second messages are cryptographically processed using public key operations and these messages have substantially the same format and cryptographic processing, and the Client and Server verify the certificate chain in the received second and first message respectively.

20

15. The method in claim 2, wherein the public key operation comprises an RSA operation or an RSA based operation.

16. The method in claim 2, wherein:

the first and second messages are created using a Signed-Inside-Enveloped-Data cryptographic primitive;

the Client-Nonce is sent to the Server encrypted by the Server's public key in the field of the public key encryption block that is normally associated with a data encryption key or with an OAEP padding seed, and this Client-nonce is used as the encryption key for the Encrypted-Data primitive, and each one contains copy of the message Sender's certificate chain;

30

the Server-Nonce is sent to the Client encrypted by the Client's public key in the field of the public key encryption block that is normally associated with a data encryption key or with an OAEP padding seed, and this Server-nonce is used as the encryption key for the Encrypted-Data primitive, and each one contains copy of the message Sender's certificate chain; and

35

transmission of said Sever-Nonce and Client-Nonce in the field normally used for a data encryption key or an OAEP padding seed enabling a single cryptographic primitive to be used for secure session setup and for secure unidirectional messaging and for other secure protocol applications.

17. The method in claim 16, wherein said cryptographic primitives for Signed-Inside-Enveloped-Data provide transport of a secret key from Sender to Recipient using a public key of the recipient.

18. The method in claim 16, wherein said single cryptographic primitive comprises a Signed-Inside-Enveloped-Data primitive.

19. The method in claim 2, wherein the Data carried in the first message is a Client-Nonce and the data carried in the second message is the Server-Nonce.

20. The method in claim 2, wherein a digitally signed portion of the second message can be pre-computed and/or reused with different messaging sessions, and so that the Server need not perform a computationally expensive private key operation to initiate a secure session.

21. The method in claim 2, wherein a digitally signed portion of the second message is pre-computed for different messaging sessions and no session specific private key operation is performed to initiate a secure session.

22. The method in claim 2, wherein a digitally signed portion of the second message is reused from an earlier session for a subsequent messaging session and no session specific private key operation is performed to initiate the subsequent secure session.

23. The method in claim 2, wherein the cryptographic transformation in the third and fourth messages are the same.

24. The method in claim 2, wherein the cryptographic transformation in the third and fourth messages are different by exchanging the roles of the Client-Nonce and the Server-Nonce.

25. The method in claim 2, wherein the cryptographic transformation is a hash of the concatenation of the client-nonce and server-nonce values.

26. The method in claim 25, wherein the hash is selected from the set consisting of MD5, SHA-1, and SHA-256.

27. The method in claim 2, wherein the cryptographic transformation is an encryption of one of either the client-nonce value or the server-nonce value using the other nonce value as the key.

28. The method in claim 27, wherein the cryptographic transformation encryption is selected from the set consisting of triple-DES, XTEA, RC5, and AES.

29. The method in claim 2, wherein the third and fourth messages are created using an Encrypted-Data cryptographic primitive, and wherein the Encrypted-Data key for the third message is different than the Encrypted-Data key for the fourth message, and both Encrypted-Data keys are derived from a Master Key that is computed with the aid of one or more applications of a cryptographic hash function applied to at least the Client-Nonce and the Server-Nonce.

30. The method in claim 29, wherein the Master Key is computed with the aid of one or more applications of a cryptographic hash function applied to the Client-Nonce and the Server-Nonce and to some or all of the information in the previously send or received messages.

31. The method in claim 30, wherein the Master Key (MK) is computed as the concatenation of at least a portion of the server-nonce, a portion of the client-nonce, and a portion of the first and second messages.

32. The method in claim 30, wherein the Master Key (MK) is computed as a concatenation as follows:  $MK = \text{HMAC}(\text{Server-Nonce} \parallel \text{Client-Nonce}, \text{SHA1}(\text{First-Message}) \parallel \text{SHA1}(\text{Second-Message}))$ .

33. The method in claim 29, wherein the Encrypted-Data key for the third message equals HMAC (MK, Client-Subject-Name), where a Client-Subject-Name is generated from one or more fields extracted from the Client's certificate.

34. The method in claim 29, wherein the Encrypted-Data key for the fourth message equals HMAC (MK, Server-Subject-Name), where Server-Subject-Name is one or more fields extracted from the Server's certificate.

35. The method in claim 29, wherein: the Encrypted-Data key for the third message equals HMAC (MK, Client-Subject-Name), where a Client-Subject-Name is generated from one or more fields extracted from the Client's certificate; and the Encrypted-Data key for the fourth message equals HMAC (MK, Server-Subject-Name), where Server-Subject-Name is one or more fields extracted from the Server's certificate.

36. A method for conducting secure interactive communication sessions between a server and a client, said method comprising:

    sending a first message containing a first token chosen by the client;

    receiving said first message including said first token by the server;

    sending a second message containing a copy of the first token extracted from the first message, and a second token that was chosen by the server, by the server;

exchanging third and fourth messages between the client and the server, said third and fourth messages including a content portion having substantially the same format and cryptographic processing as each other, the contents portions of the third and fourth messages including a cryptographic transformation of at least the first token and second token; and

- 5 each of the server and client examining the respective received third and fourth messages to confirm that they were created by an entity that knew both the first token and the second token.

37. The method in claim 36, wherein the cryptographic transformation is slightly different in the third and fourth messages.

10 38. The method in claim 36, wherein the first token comprises a client-nonce and the second token comprises a server-nonce.

15 39. A computer program product for use in conjunction with a computer system having a server and a client, the computer program product comprising a computer readable storage medium and a computer program mechanism embedded therein, the computer program mechanism, comprising: a program module that directs the computer system and/or components thereof including at least one of the client or server, to function in a specified manner to conduct secure interactive communication sessions between a server and a client, the communications occurring in a computer system hardware architecture neutral and operating system neutral and network transport protocol neutral manner for secure interactive communication sessions, the program module including instructions for:

20 sending a first message containing a first token chosen by the client;

receiving the first message including the first token by the server;

25 sending a second message containing a copy of the first token extracted from the first message, and a second token that was chosen by the server, by the server;

30 exchanging third and fourth messages between the client and the server, the third and fourth messages including a content portion having substantially the same format and cryptographic processing as each other, the contents portions of the third and fourth messages including a cryptographic transformation of at least the first token and second token; and

each of the server and client examining the respective received third and fourth messages to confirm that they were created by an entity that knew both the first token and the second token.

40. The computer program in embodiment 39, wherein the cryptographic transformation is slightly different in the third and fourth messages.